

セグメント分解法による高速図形発生方式とその評価†

小林 芳樹^{††} 高藤 政雄^{††} 浜田 長晴^{††}
 平沢 宏太郎^{††} 川本 幸雄^{††}
 小中 清司^{†††} 桑原 洋^{†††}

近年の多様化し、膨大化する情報を扱うため CAD/CAM 等の分野では高速なカラーグラフィック・ディスプレイが望まれてきている。しかし従来、図形発生のために1ドット単位に画像メモリに書き込みを行っており、これゆえに図形発生速度が制限されていた。これに対し我々は、図形を水平あるいは垂直成分のセグメントに分解して画像メモリに書き込むセグメント分解 (SGD) 方式を提案し、直線のセグメント分解アルゴリズムおよび図形発生装置の構成を示した。次に、SGD 方式の図形描画処理速度の分析、評価を行い、画像メモリに水平方向に16ドットあるいは垂直方向に8~16ドット同時に書き込めるようにした場合、

- (1) 直線描画速度を従来方式の2.2倍にできる
- (2) 図形中に縦横線の占める割合が大きくなるほど高速化の効果が大きく、たとえば縦横線が40%の場合、従来方式の3.2~3.5倍にできることを明らかにした。

1. ま え が き

ラスタスキャン形グラフィック・ディスプレイ端末は色彩図形の表示が可能であるという特徴を持ち、また図形発生速度も年々高速になってきている。我々も主にプロセス制御用途を対象に高機能、高速応答および使い易さを追求したカラーグラフィック・ディスプレイ^{1),2)}を開発した。本ディスプレイは7,000本/秒(100ドット/本)の直線描画速度を持つが、近年の多様化、膨大化する情報を扱う CAD/CAM (Computer Aided Design/Manufacturing) 分野等ではより高速な図形発生が望まれてきている。

画像メモリを持つ形式のグラフィック・ディスプレイでは、従来その図形の発生のために1ドット単位に画像メモリに書き込みを行っており、これゆえに図形発生速度が制限されていた。これに対し我々は更に高速化を図るため、図形を水平あるいは垂直成分のセグメントに分解してこのセグメント単位に画像メモリに書き込む、新しいセグメント分解 (Segment Decomposition) 法による高速図形発生方式を提案するものである。本稿では、セグメント分解方式による直線発生アルゴリズムおよび図形発生装置の構成について述べ、本方式

による図形描画処理時間の分析および評価を行う。なお、従来方式による直線、円等の発生アルゴリズムを扱った文献^{3),4)}は多いが、本稿のようなセグメント分解方式の分析、評価を行ったものは見当たらない。

2. グラフィック・ディスプレイと図形表示速度

2.1 ラスタスキャン形ディスプレイ

ラスタスキャン形グラフィック・ディスプレイは図1

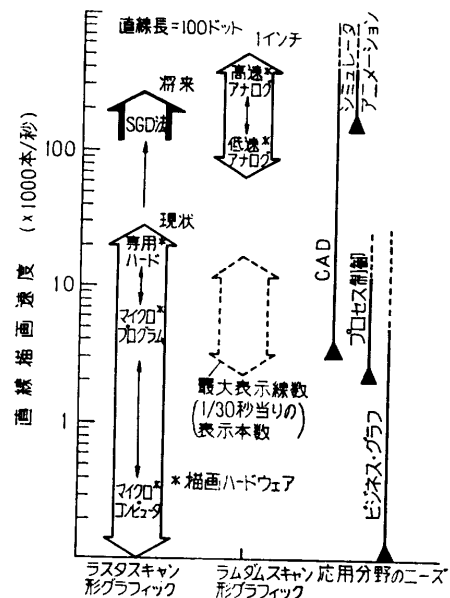


図1 グラフィック・ディスプレイの図形発生速度と応用分野

Fig. 1 Picture generating speed of graphic displays and application needs.

† A High Speed Picture Generation Method by Segment Decomposition of Pictures by YOSHIKI KOBAYASHI, MASAO TAKATOO, NAGAHARU HAMADA, KOOTARO HIRASAWA, YUKIO KAWAMOTO (Hitachi Research Laboratory, Hitachi Ltd.), KIYOSHI KONAKA and HIROSHI KUWAHARA (Omika Works, Hitachi Ltd.).

†† (株)日立製作所日立研究所
 ††† (株)日立製作所大みか工場

に示すように、その図形発生速度がマイクロコンのプログラムによる100本/秒程度のものからハードウェアによる2万本/秒程度のものまで各種のレベルのものがあり、低価格で色彩図形の表示が可能ということで広い分野に用いられてきている。しかし、高速な線図形の表示を特徴とするランダムスキャン形のものに比較すると直線描画速度は1桁以上遅く、高速なレスポンスや滑らかな動きを必要とするCADや、リアルタイムの動きが要求されるシミュレータ等の分野に適用することは困難であった。我々は主にプロセス制御用途に、マイクロプログラムにより図形発生を行う直線描画速度7,000本/秒のラスタスキャン形グラフィックディスプレイ^{1),2)}を開発したが、前述のような

より多くの応用分野に適用するためにはランダムスキャン形並みの図形発生速度の達成が必要であり、このための高速化手段を次節にて検討する。

なお本稿での図形とは、文字や直線、円、円弧、面等の幾何学的図形を指しており、ITVカメラ等からの任意画像は含まない。

2.2 図形発生処理の高速化手段と効果

グラフィック端末で図形を管理し、図形の拡大、回転等の操作を行えるようなシステムでは、図形コマンドは、図2(a)に示すように、

① 座標変換処理……ユーザ定義の座標系での直線、円等の図形を、その移動、拡大、回転等の操作情報およびCRT画面への表示制御情報に基づいて座標変換しCRT画面座標系に変換する処理

② クリッピング処理……上記処理で得られたCRT画面座標系での図形のCRT画面外に出た部分を切り取る処理

③ ドット展開処理……上記処理で得られたCRT画面座標の図形を最小表示単位のドット(絵素)に展開して画像メモリに書込む処理

により表示される。前述のディスプレイ^{1),2)}では、小規模のハードウェアで高速な図形発生を達成するため、①~③の処理をグラフィック・プロセッサのマイクロプログラムで実行するようなシステムとした。本ディスプレイでの処理時間は図3(a)に示すようになっており、座標変換、クリッピング処理がない場合には約7,000本/秒(100ドット/本)の直線描画速度である。

そこで図形発生をより高速にするために、図2(b)に示すように、①~③の個々の処理の高速化を図るとともに、これらの処理を独立したモジュールとし、出力結果(座標データ)を次段モジュールで直ちに処理するパイプライン制御による並列処理を行うことが考えられる。パイプライン制御では一番長い処理時間のモジュールにその処理速度が制限されるが、座標変換処理、クリッピング処理は専用ハードウェア^{5),7)}により数μ秒のオーダーにまで高速化できる。したがって③のドット展開処理の高速化が問題になるわけであるが、次の2面から高速化を検討すると、

(1) 画像メモリへの書込回数の削減……従来図形を1ドットごとに画像メモリに書込んで

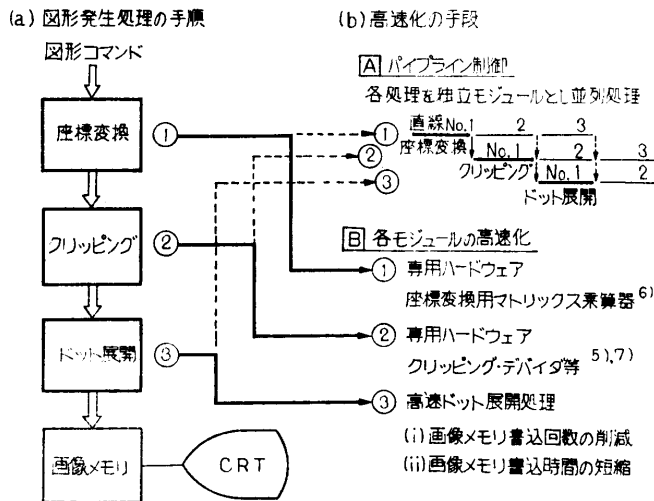


図2 図形発生処理手順と高速化の手段

Fig. 2 A process of picture generation and means of high-speed processing.

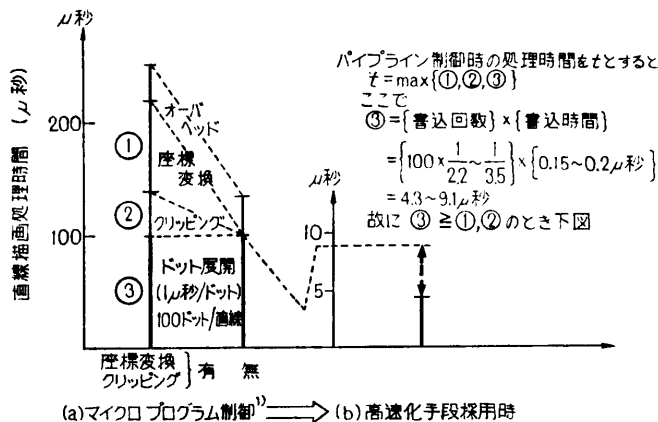


図3 直線描画処理の高速化

Fig. 3 High-speed processing of a straight line.

いたが、できる限り多くのドットを一度に書込むことにより高速化を図る。これによる高速化の程度を次章以下に評価するが、約2.2倍、画面の表示内容によっては3.5倍以上に高速化できる。

(2) 画像メモリへの書込時間の短縮……これは画像メモリに用いるメモリ素子に依存するが、我々の開発したディスプレイでは16KビットのDynamic RAMを用い、1 μ 秒をCRT表示用読出しと図形発生用書込との時分割制御していた。これに対し近年の大容量、高速RAM(たとえば日立C-MOS 16KビットStatic RAM等)を使用し、図形書込優先にすると0.15~0.2 μ 秒、すなわち5~7倍程度の高速化が可能である。

ことがわかり、図3(b)に示すように10~25万本/秒の直線描画速度の達成が可能となる。

上述のようにランダムスキャン形並の性能を持ったラスタスキャンカラーグラフィック・ディスプレイ実現への見通しを得たが、本稿では、特に画像メモリへの書込回数を削減する方式について検討し、評価した結果について述べる。

3. 高速図形発生方式の立案

本稿で述べる高速図形発生方式とは、

『図形の水平または垂直方向に連続するセグメントを、できる限り一度に画像メモリに書込む』ようにしたもので、セグメント分解(SGD…Segment Decomposition)方式と呼ぶ。本章では先ず直線のセグメント分解アルゴリズムについて述べ、次にSGD方式によるグラフィック・プロセッサの構成について簡単に言及する。

3.1 直線のセグメント分解アルゴリズム

ここでは基本図形である直線のセグメント分解処理について述べる。SGD方式では図形を水平または垂直方向に連続するセグメントに分解する必要がある、

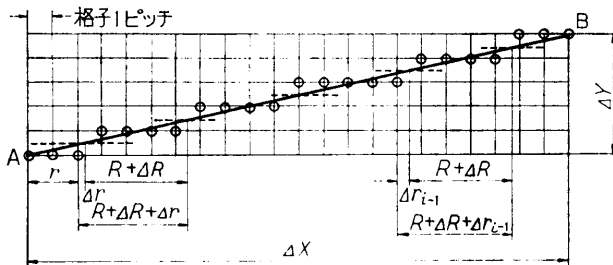


図4 直線(第1オクタント)の水平セグメントへの分解
Fig. 4 Segment decomposition of a straight line.

そのアルゴリズムを図4に示す傾き $0^\circ \sim 45^\circ$ の第1オクタントの直線 \overrightarrow{AB} を例にとり説明する。

直線 AB は第1オクタント($\Delta X > \Delta Y > 0$)であり、したがって水平(X)方向に連続するセグメントに分解される。そのセグメントは垂直(Y)方向の格子の1/2点で直線を分割しその分割線分の水平方向長を量子化することで得られる、そこで

$$\Delta X / \Delta Y = R + \Delta R \quad (\text{ただし } R: \text{整数}, 0 \leq \Delta R < 1)$$

$$\Delta X / 2\Delta Y = r + \Delta r \quad (\text{ただし } r: \text{整数}, 0 \leq \Delta r < 1)$$

と $R, \Delta R, r, \Delta r$ を置くと、最初のセグメント長(セグメントに含まれるドット数)は $r+1$ となり、 Δr が量子化残余となる。次のセグメント長は $(R + \Delta R + \Delta r)$ の整数部となり、その小数部が次のセグメントへの残余となる。

さて第 $i-1$ 番目のセグメントの残余(Δr_{i-1} とする)が得られたとき、第 i 番目のセグメント長は

$$R + \Delta R + \Delta r_{i-1}$$

の整数部となり、小数部が第 $i+1$ 番目のセグメントへの残余となる。ここで $R, \Delta R, \Delta r_{i-1}$ は

$$R: \text{整数}, 0 \leq \Delta R < 1, 0 \leq \Delta r_{i-1} < 1$$

であるため第 i 番目のセグメント長は R か $R+1$ のどちらかとなる。すなわち、

(1) $\Delta R + \Delta r_{i-1} < 1$ ならばセグメント長 R 、次のセグメントへの残余 $\Delta r_i = \Delta R + \Delta r_{i-1}$

(2) $\Delta R + \Delta r_{i-1} \geq 1$ ならばセグメント長 $R+1$ 、次のセグメントへの残余 $\Delta r_i = \Delta R + \Delta r_{i-1}$

となる。したがって判別式を $D_i = \Delta r_{i-1} - 1$ と置くことにより図5に示すアルゴリズムでセグメントに分解することができる。ほかのオクタントの直線も同様に水平または垂直のセグメントに分解することができる。

3.2 SGD方式図形発生装置の構成

SGD方式のグラフィック・ディスプレイの図形発生装置は、図6に示されるモジュールから構成される。ここではセグメント分解のために

(1) グラフィック・プロセッサ……与えられた図形コマンドを座標変換、クリッピング処理してCRT画面上の座標とし、この図形を図5のInitial Routineに示す前処理を行ってSGDプロセッサに起動を与えるプロセッサ。前章で述べたように、これらの一連の処理は高速ハードウェアによりパイプラインで制御される。

(2) SGDプロセッサ……グラフィック・プロセッサの指令に基づき、図5のSGD Main

```

Initial Routine:
  Calculate R, ΔR, r, Δr from ΔX, ΔY
  (Xs, Ys)=(XA, YA) /* (Xs, Ys) is a start point */
  Di=Δr-1 /* to draw a segment */
End
1st Octant SGD Main Routine:
  Draw a 1st segment with r+1 dots; Xs=Xs+r;
  Ys=Ys+1
  Do while Ys<Yb
    Di=Di+ΔR
    if Di≥0 then draw a segment with (R+1) dots
      Di=Di-1
      Xs=Xs+R+1
    else draw a segment with R dots
      Xs=Xs+R
  Ys=Ys+1
  End
  if Δr=0 then draw a last segment with r dots
  else draw a last segment with r+1 dots
  End
  
```

図5 直線のセグメント分解アルゴリズム

Fig. 5 An algorithm for segment decomposition of a straight line.

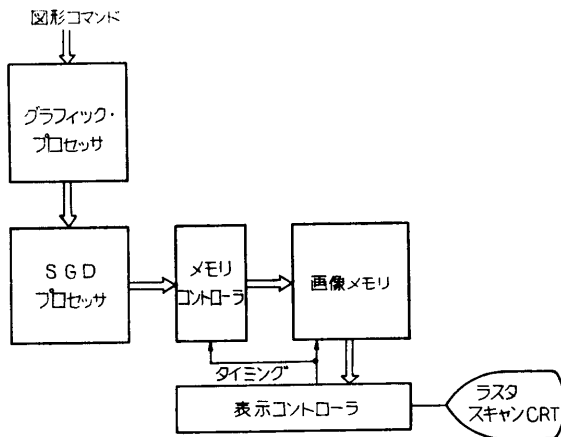


図6 SGD方式の図形発生装置の構成

Fig. 6 Block diagram of a graphics generating units adopting SGD method.

Routine に示される処理を高速に行うプロセッサ.

(3) メモリコントローラ……SGD プロセッサから与えられたセグメント情報をもとに、画像メモリの書込単位に合うように(場合によっては複数回に書込みを分割)して書込みを実行する制御回路. のように機能分担する構成を検討している.

さて、与えられたセグメントを画像メモリに一度に書込めるか否かは画像メモリの構成に依存し、これが性能に影響する. そこで最適な画像メモリの構成を検討するために、SGD方式を画像メモリの書込み上の制約から.

(1) SGD-H (Horizontal)……水平方向にだけ l ドット同時に書込める

(2) SGD-HV (Horizontal/Vertical)……水平方向に l ドット同時に、あるいは垂直方向に m ドット同時に書込める.

に分類し、描画処理速度を評価することにした. なお水平 l ドット \times 垂直 m ドットのセルへ直線を分解する方式も考えられるが、この方式ではセルに高速に分解するハードウェアが複雑になりすぎる. このため我々は結論的には SGD-HV 方式 ($l=16$ ドット, $m=8\sim 16$ ドット) が性能/価格比の点で最も優れていると評価している.

4. 図形描画速度(書込回数)の分析

前章で述べたように図形は水平または垂直方向に連続するセグメント単位で書込まれる. 図形描画速度は画像メモリへの書込回数と書込時間により決まる. ここでは、どのような方式をとにしても書込時間は同じものとし、したがって図形描画速度を各方式における画像メモリの書込回数で評価することにした. 本章で書込回数の分析を行い、次章で各方式の評価を行う.

なお、画像メモリには画面格子(ドット)に量子化して書込むわけであり書込回数を量子化されたデジタル量として扱うべきであるが、本章の分析手法では多くの図形発生における平均という意味で、たとえば書込みセグメント長等のようにアナログ量で扱っている部分もある. このため、直線が短い場合には誤差が出る予想されるが、そうでない場合には平均的な書込回数になる.

4.1 直線の分析

図7に示すように、第1オクタントの勾配 θ 、長さ L の直線の総ドット数 $N_{d\theta}$ はその長辺の長さ $L \cos \theta$ となる. したがって θ が一様に分布しているとする. 平均的な $N_{d\theta}$ は

$$N_{d\theta} = \int_0^{\pi/4} L \cos \theta d\theta \Big/ \frac{\pi}{4} = \frac{2\sqrt{2}}{\pi} L \dots\dots\dots (1)$$

* 角度 $0\sim\pi/4$ を微小角 $\Delta\theta$ に n 分割したとき、分割直線の長辺の長さ $L \cos(k \cdot \Delta\theta)$, $k=0\sim n-1$ の平均をとり、 $n\rightarrow\infty$ にして平均の $N_{d\theta}$ を求めている. すなわち、

$$\begin{aligned}
 N_{d\theta} &= \lim_{n\rightarrow\infty} \frac{\sum_{k=0}^{n-1} L \cos(k \cdot \Delta\theta)}{n} \\
 &= \lim_{n\rightarrow\infty} \sum_{k=0}^{n-1} L \cos(k \cdot \Delta\theta) \Delta\theta / n \cdot \Delta\theta \\
 &= \int_0^{\pi/4} L \cos \theta d\theta \Big/ \frac{\pi}{4} \quad (\because n \cdot \Delta\theta = \frac{\pi}{4})
 \end{aligned}$$

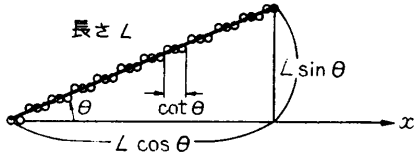


図7 直線を構成するドットの分析
Fig. 7 Analysis of dots in a straight line.

となる。一方直線の勾配が θ の場合、水平方向に連続するセグメントの数 N_{lv} はその短辺の長さ $L \sin \theta$ となるので、平均的な N_{lv} は

$$N_{lv} = \int_0^{\pi/4} L \sin \theta d\theta \Big/ \frac{\pi}{4} = \frac{2(2-\sqrt{2})}{\pi} L \quad \dots (2)$$

となる。

さて画像メモリの水平方向に一度に書込めるドット数を l とする。なお、これは任意の座標から l ドット一度に書込めるのではなく、メモリ素子のアドレス付けの制限により規則的に区切られた l ドット区間内であれば一度に書込めるような、定まった区切りのついた長さ l である。今、水平方向長 $L \cos \theta$ の直線を水平区分長 l で区切ったときの平均区切り数は $L \cos \theta / l$ であり、一方、勾配 θ の直線の平均水平セグメント長は $\Delta X / \Delta Y = \cot \theta$ である。また長さ $\cot \theta$ の水平セグメントが連続して並べられているとき、ある1つの区切りがセグメントを2つに分割する確率は $(\cot \theta - 1) / \cot \theta$ である。ゆえに、水平方向 l ドットの区切りによりセグメントが分割される数 ΔN_{lv} の平均は、

$$\begin{aligned} \Delta N_{lv}(l) &= \int_0^{\pi/4} (L \cos \theta / l) \cdot \{(\cot \theta - 1) / \cot \theta\} d\theta \Big/ \frac{\pi}{4} \\ &= \frac{4(\sqrt{2}-1)}{\pi} \cdot \frac{L}{l} \quad \dots (3) \end{aligned}$$

となる。

したがって、一般に長さ L の直線の平均的な書込回数 N_v は、画像メモリに水平方向 l ドット、垂直方向 m ドット一度に書込める場合には

$$\begin{aligned} N_v &= N_{lv} + \{\Delta N_{lv}(l) + \Delta N_{lv}(m)\} / 2 \\ &= \frac{2}{\pi} \left\{ 2 - \sqrt{2} + (\sqrt{2} - 1) \left(\frac{1}{l} + \frac{1}{m} \right) \right\} L \quad \dots (4) \end{aligned}$$

となる。

4.2 円の分析

図8(a)に示すように、半径 R の円の $1/8$ 円の部分に着目すると、この部分の総ドット数 N_{dc} は、

$$N_{dc} = \frac{\sqrt{2}}{2} R \quad \dots (5)$$

となる。一方水平方向に連続するセグメント数 N_{lc} は

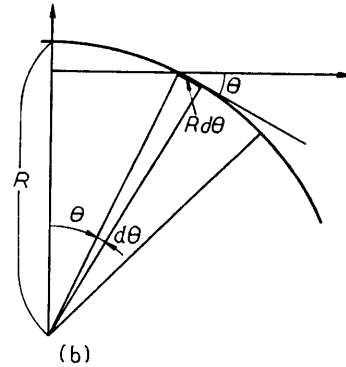
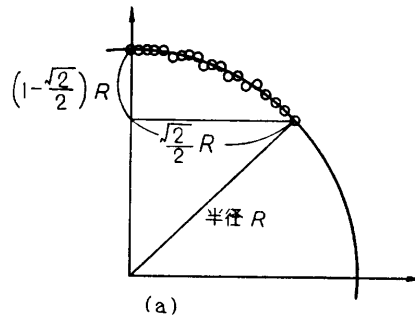


図8 円を構成するドットの分析
Fig. 8 Analysis of dots in a circle.

$$N_{lc} = \left(1 - \frac{\sqrt{2}}{2} \right) R \quad \dots (6)$$

となる。さて、図8(b)のように角度 θ における微小角度 $d\theta$ における円弧を考えると、これは長さ $R d\theta$ 、水平方向長 $R d\theta \cos \theta$ の微小線分と見做せる。この線分の平均水平セグメント長は $\cot \theta$ であり、水平方向 l ドットの区切りによる平均区切り数は $R d\theta \cos \theta / l$ で、1つの区切りがセグメントを2つに分割する確率は $(\cot \theta - 1) / \cot \theta$ である。ゆえに、水平方向に一度に l ドット書込める場合に、 $1/8$ 円の水平セグメントが2つに分割される数 ΔN_{lc} は、

$$\Delta N_{lc}(l) = \int_0^{\pi/4} \{(\cot \theta - 1) / \cot \theta\} (R \cos \theta / l) d\theta \quad \dots (7)$$

となる。

したがって、一般に半径 R の円の平均的な書込回数 N_c は、画像メモリに水平方向 l ドット、垂直方向 m ドット一度に書込める場合には、

$$\begin{aligned} N_c &= 8N_{lc} + 4\Delta N_{lc}(l) + 4\Delta N_{lc}(m) \\ &= 4 \left\{ 2 - \sqrt{2} + (\sqrt{2} - 1) \left(\frac{1}{l} + \frac{1}{m} \right) \right\} R \quad \dots (8) \end{aligned}$$

となる。

4.3 面の分析

水平方向の長さ L 、垂直方向の長さ M の矩形の面図

形を発生することを考える。面図形を発生するための総ドット数 N_d は

$$N_d = L \cdot M \dots\dots\dots (9)$$

となる。一方水平方向に連続するセグメント長は L であり、画像メモリの水平方向に一度に l ドット書込めるとすると、これによる水平区切り数は L/l である。したがって $L \times M$ の面を発生するための書込回数 N_s は

$$N_s = (1 + L/l)M \dots\dots\dots (10)$$

となる。

5. 図形描画速度の評価

ここでは前章の分析結果を基に、画像メモリへの書込時間を一定とした場合、従来の1ドットごとの書込方式に対してSGD方式がどの程度高速になっているかを評価する。なお評価のパラメータとしては、

(1) SGD-H, SGD-HV方式における水平、垂直方向に一度に書込めるドット数 l, m

(2) 表示図形中に縦横線が占める割合 r ……応用分野にもよるが、通常の画面中の直線では縦横線の割合が高く、この割合 r をパラメータとした。なお縦線と横線の割合は同一で、各々 $r/2$ とする。をとった。

5.1 直線の評価

従来方式では、縦横線の割合が r の場合、長さ L の直線を描画するための書込回数は、平均すると(1)式より、

$$r \cdot L + (1-r) \frac{2\sqrt{2}}{\pi} L \dots\dots\dots (11)$$

となる。一方SGD-HV方式での書込回数は、(4)式より、

$$\frac{r}{2} \cdot \frac{L}{l} + \frac{r}{2} \frac{L}{m} + (1-r) \frac{2}{\pi} \left\{ 2 - \sqrt{2} + (\sqrt{2} - 1) \times \left(\frac{1}{l} + \frac{1}{m} \right) \right\} L \dots\dots\dots (12)$$

となる。したがってSGD-HV方式の従来方式に対する直線描画処理速度の比 E_v は

$$E_v = \frac{r + (1-r) \frac{2\sqrt{2}}{\pi}}{\frac{r}{2} \left(\frac{1}{l} + \frac{1}{m} \right) + (1-r) \frac{2}{\pi}} \times \left\{ 2 - \sqrt{2} + (\sqrt{2} - 1) \left(\frac{1}{l} + \frac{1}{m} \right) \right\} \dots\dots\dots (13)$$

となる。この処理速度比 E_v を l, m, r をパラメータとしたグラフを図9~12に示す。このグラフから

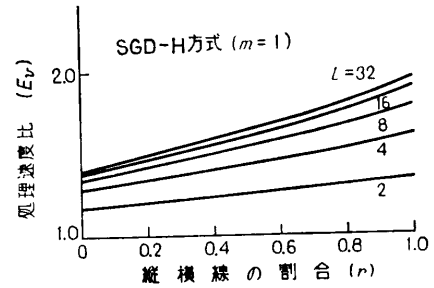


図9 直線描画の処理速度比 [SGD-H方式]
Fig. 9 Drawing speed ratio of straight lines [SGD-H Method].

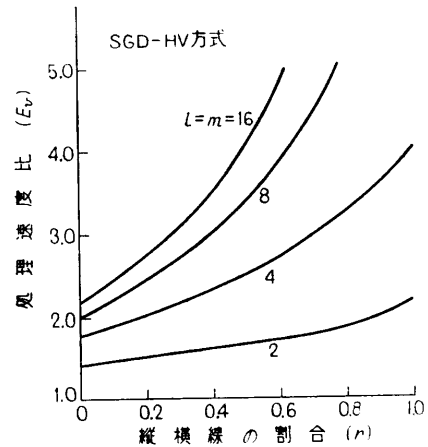


図10 直線描画の処理速度比 [SGD-HV方式]
Fig. 10 Drawing speed ratio of straight lines [SGD-HV Method].

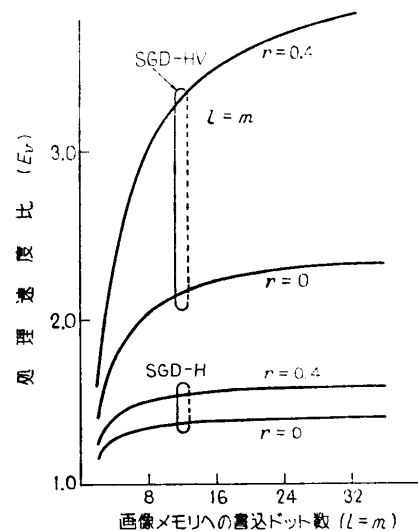


図11 書込ドット数と直線描画の処理速度比 [1]
Fig. 11 Relation between l, m and drawing speed ratio of straight lines [1].

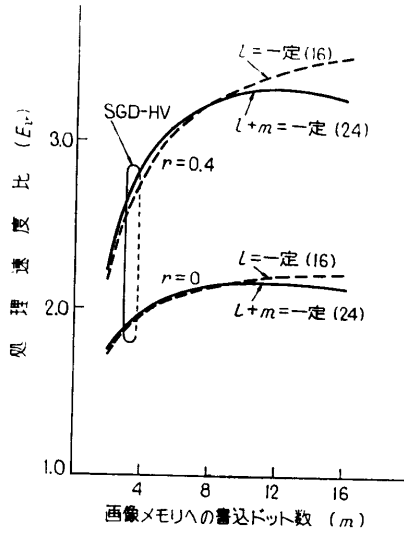


図 12 書込ドット数と直線描画の処理速度比 (2)
Fig. 12 Relation between l, m and drawing speed ratio of straight lines (2).

(1) 縦横線の割合 r の効果

① SGD-H ($l=32$) 方式では, $r=0$ (直線の勾配がランダム) のとき $E_s=1.4$ で, r が大きくなっても E_s はそれほど大きくなりず 2 以下である。

② SGD-HV ($l=m=16$) 方式では, $r=0$ のとき $E_s=2.2$ で, r が大きくなると E_s も大きくなり $l(=16)$ に近づく。

(2) 画像メモリへの書込ドット数 l, m の効果

① SGD-H 方式では, r の値にかかわらず, $l=8$ で $l=\infty$ の場合の E_s の 80% (ベースは $E_s=1$ とする), $l=16$ で 90% を超える。

② SGD-HV 方式では,

(i) $l=m$ とした場合, $r=0$ のとき $l=m=10$ で $l=m=\infty$ の E_s の 80% を超え, $r=0.4$ のとき $l=m=16$ で 80% を超える。 r が大きくなるほど, $l=m$ を大きくする効果が大きくなる。

(ii) $l+m=一定$ とした場合, (13) 式から明らかのように $l=m$ に近づくほど E_s は大きくなる。

ことがわかる。この結果, SGD-H 方式では縦線を従来方式と同様に 1 ドットごとに書込んでおりそれほど性能が向上しないため, SGD-HV 方式が望ましいことがわかる。SGD-HV 方式では, 画像メモリの現状のハードウェア技術と性能のバランスからみて, $l=16, m=8\sim 16$ 程度が最適と考えられる。このとき $E_s=2.2$ ($r=0$), $E_s=3.2\sim 3.5$ ($r=0.4$) となる。

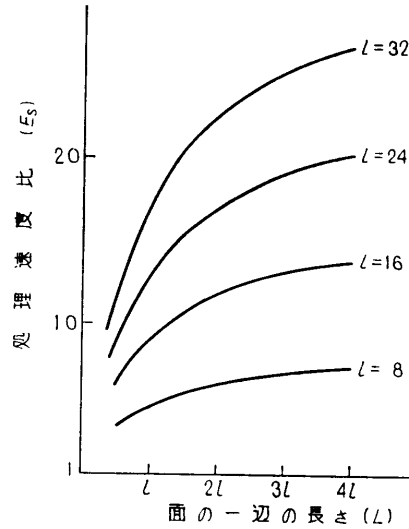


図 13 面描画の処理速度比
Fig. 13 Drawing speed ratio of rectangles.

5.2 円の評価

従来方式では半径 R の円を描画するための書込回数は, 平均すると (5) 式より $4\sqrt{2}R$ であり, 一方 SGD-HV 方式での書込回数は (8) 式により示される。したがって SGD-HV 方式の従来方式に対する円描画処理速度の比 E_s は

$$E_s = \frac{\sqrt{2}}{2 - \sqrt{2} + (\sqrt{2} - 1) \left(\frac{1}{l} + \frac{1}{m} \right)} \dots\dots\dots (14)$$

となる。これは (13) 式の直線描画の処理速度比 E_s において $r=0$, すなわち直線の勾配がランダムであるとした場合に一致する。ゆえに SGD-HV 方式で $l=m=16$ としたとき $E_s=2.2$ となる。

5.3 面の評価

面を描画する場合の従来方式に対する処理速度比 E_s は, (9), (10) 式より

$$E_s = \frac{1}{\frac{1}{l} + \frac{1}{L}} \dots\dots\dots (15)$$

となる。これを図 13 に示すが, SGD-H, HV 方式ともに, $l=16$ の場合, 一辺 (L) が 50 ドットの矩形面であれば $E_s=12$ であり, $L \gg l$ になると $E_s \approx l$ となる。

6. むすび

高速な図形発生のために, 図形を水平あるいは垂直成分のセグメントに分解して画像メモリに書込む SGD 方式を提案し, 直線のセグメント分解アルゴリズム

ムおよび図形発生装置の構成を示した。そして SGD 方式の図形描画処理時間の分析、評価を行い、SGD-HV 方式で $l=16$, $m=8\sim 16$ の場合

- (1) 直線描画速度を従来方式の 2.2 倍にできる
 - (2) 図形中に縦横線の占める割合が大きくなるほど高速化の効果が大きく、たとえば縦横線が 40% の場合、従来方式の 3.2~3.5 倍にできる。
- ことを明らかにした。

本稿の SGD 方式はその評価が終った段階であるが、システムとして高速な図形描画速度を達成するためには画像メモリへの書込時間の短縮、グラフィック・プロセッサにおける座標変換、クリッピング、SGD 前処理の高速化が重要であり、今後これらを検討してシステムとしてまとめていきたいと考えている。

最後に、本研究の遂行にあたり日頃ご指導いただく日立製作所大みか工場森田副工場長、同日立研究所高砂所長、および第 82 研究室諸氏に深謝いたします。

参 考 文 献

- 1) 小林芳樹他：高機能ディスプレイ端末のための

グラフィック・プロセッサの開発およびその評価、情報処理学会論文誌, Vol. 22, No. 2, pp. 129-138 (1981).

- 2) Takatoo, M. et al.: Software Architecture of a Graphic Display Terminal with Interactive Picture Generation Capability, 6th IFAC/IFIP Int. Conference (1980).
- 3) Suenaga, Y. et al.: A High-Speed Algorithm for the Generation of Straight Lines and Circular Arcs, IE³ Transactions on Computer, Vol. C-28, No. 10, pp. 728-736 (1979).
- 4) Maxwell, P.C. et al.: The Generation of Polygons Representing Circles, Ellipses and Hyperbolas, Computer Graphics and Image Processing, Vol. 8, No. 10, pp. 84-93 (1979).
- 5) Sutherland, I.E.: A clipping divider, FJCC, pp. 765-775 (1968).
- 6) Hagan, T.G. et al.: The Adage Graphics Terminal, FJCC, pp. 747-755 (1968).
- 7) Clark, J.: A VLSI Geometry Processor for Graphics, COMPUTER, Vol. 13, No. 7, pp. 59-68 (1980).

(昭和 55 年 10 月 8 日受付)

(昭和 56 年 3 月 19 日採録)